Draft Model Knows When to Stop: A Self-Verification Length Policy for Speculative Decoding

Ziyin Zhang[1,2*] **Jiahao Xu**[2] **Tian Liang**[2] **Xingyu Chen**[1,2*] **Zhiwei He**[1,2*]
Rui Wang[1†] **Zhaopeng Tu**[2]
[1]Shanghai Jiao Tong University
[2]Tencent AI Lab
[1]{daenerystargaryen,galaxychen,zwhe.cs,wangrui12}@sjtu.edu.cn
[2]{jettexu,ttianliang,zptu}@tencent.com

Abstract

Speculative Decoding (SD) has become an important technique in accelerating the inference speed of large language models. Conventional SD methods employ a fixed draft length, which ignores the token generation difficulty across tasks. Consequently, in this paper, we address such an issue and introduce SVIP - a difficulty-aware dynamic draft length policy for speculative decoding systems. Based on a theoretical lower bound of draft token acceptance rate and its inference-time approximation, SVIP adaptively determines the lengths of draft sequences based on the entropy of each draft token distribution. Experimental results on mainstream SD benchmarks and frameworks demonstrate the superior performance of SVIP, achieving up to 20% walltime speedup on SpecBench over baseline SD methods and 60% speedup on MT-Bench for long-form generation of up to 8K tokens. Moreover, SVIP is totally training-free and compatible with any existing SD methods that generate draft tokens autoregressively. Experimental results also show that SVIP yields consistent walltime improvement on top of GliDe & CaPE and EAGLE-2.

1 Introduction

Speculative decoding (Leviathan et al., 2023; Chen et al., 2023) is a novel technique that markedly enhances the generation wall-time of large language models (LLMs). This approach employs a small and efficient amateur model to draft sequences, while concurrently utilizing a larger and more powerful expert model to verify the drafts. By avoiding the autoregressive generation of each token through the target LLM, speculative decoding achieves improved efficiency while preserving the quality of the output.

Many variants of speculative decoding have been proposed. A line of work focuses on developing a stronger and faster draft model (Li et al., 2024b; Elhoushi et al., 2024; Du et al., 2024). Another line of work contributes to maximizing the acceptance probability of draft tokens (Sun et al., 2023; Li et al., 2024a; Lu et al., 2024). In general, they all tend to maximize the alignment between the draft and target model to further maximize the system acceptance rate.

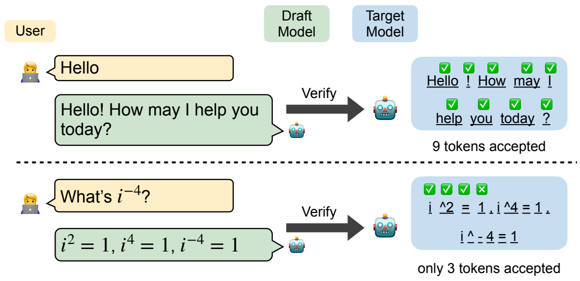

Figure 1: The "difficulty" of tokens varies in a sequence, resulting in different numbers of accepted draft tokens at different positions.

Though successful, most of these works limit their settings to a fixed draft length, where the draft model always generates a fixed number of tokens (e.g. 4 or 5) in each iteration. Such a setting ignores the fact that some tokens - such as stop words or civilities - in the generation may be easy for the draft model to predict, while others - such as knowledge-intensive or reasoning-intensive tokens - can be much harder, as shown in Figure 1.

To address this issue, in this work we introduce SVIP - Self-VerIfication length Policy, a simple, plug-and-play dynamic draft length policy for speculative decoding systems, which enhances the wall time speedup of these systems by adaptively allowing for longer draft sequences for "simple" tokens (top of Figure 1) and terminating the drafting process early upon encountering "hard" tokens (bottom of Figure 1).

Specifically, we first analyze the acceptance rate

*Work done during their internship at Tencent AI Lab.
†Corresponding author.

Preprint — https://doi.org/10.32388/I3VW8J

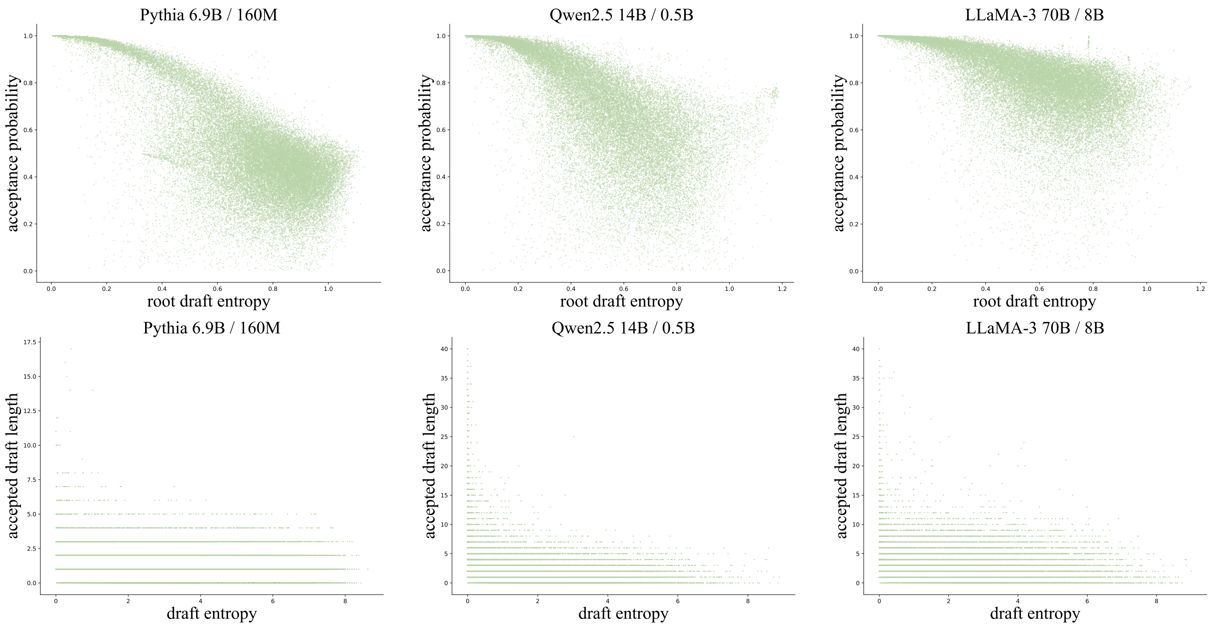

Figure 2: The correlation between draft model entropy and draft token acceptance probability (top) and lengths of accepted draft seqeunces (bottom).

of speculative decoding systems and derive a lower bound based on the system's entropy information. Further empirical analysis suggests that such a bound can be approximated by the entropy of the draft model only, which is naturally available in the drafting process of any auto-regressive draft models. Consequently, we develop SVIP which controls the length of draft sequences dynamically by determining whether to continue drafting or start verification upon the generation of each draft token.

With extensive experiments across multiple model sizes and generation lengths, we demonstrate the superior performance of SVIP. It yields more than 20% of improvements over vanilla speculative decoding for Qwen2.5 14B and LLaMA-3 70B on SpecBench (Xia et al., 2024), and more than 60% of improvements for Pythia 6.9B on MT-Bench (Zheng et al., 2023) when generating long-form responses of up to 8K tokens.

Moreover, since our method is lightweight and training-free, it is extremely flexible and can be adapted to any speculative decoding system with an auto-regressive draft model. As examples, we apply SVIP on top of two state-of-the-art speculative decoding methods: GliDe with a CaPE (Du et al., 2024) and EAGLE-2 (Li et al., 2024a), and confirm that it brings consistent improvements. Our code is available at `https://github.com/Geralt-Targaryen/SVIP`.

In summary, our contributions are threefold:

1. We derive a low bound of speculative decoding systems, where the acceptance rate of the draft model could be modeled by its entropy only.

2. Based on this lower bound, we further develop an entropy-based dynamic draft length policy for speculative decoding systems, which is extremely flexible and can be adapted to any auto-regressive draft model.

3. Experimental results demonstrate the superior performance of SVIP over baseline methods, with up to 20% average speedup on SpecBench, 60% in long-form generation, and consistent improvement over state-of-the-art speculative decoding frameworks such as GliDe & CaPE and EAGLE-2.

2 Method

The overall objective of SVIP is to dynamically adapt draft length on-the-fly, stopping early if the current draft token's acceptance probability is low and otherwise continuing drafting. To introduce our method, we first provide the background on speculative decoding in Section 2.1. Then, based on the key observation that the acceptance probability of a draft token depends on the target model confidence - which is unavailable in the drafting phase, we derive a theoretical lower bound for the acceptance rate based on the draft model's entropy

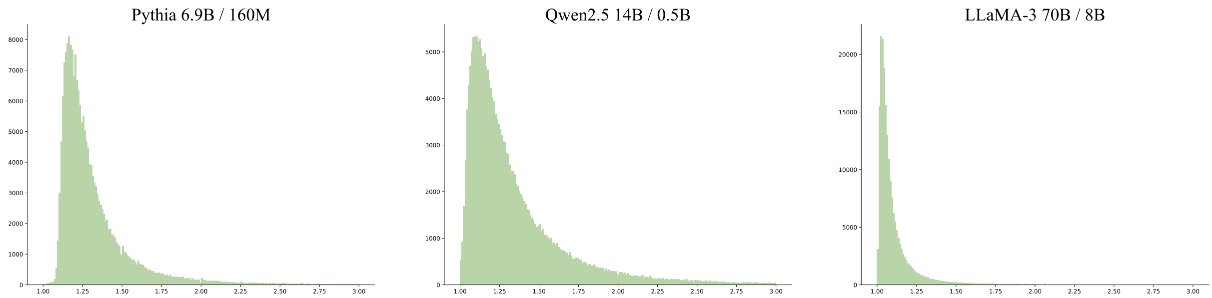

Figure 3: Distribution histograms of the entropy ratio $H_{q,p}/H_q$. For most tokens, this ratio falls into a narrow range, indicating that the cross entropy $H_{q,p}$ can be approximated by a constant multiplication of the draft entropy H_q.

and the cross-entropy between the target and draft models in Section 2.2. Finally, based on empirical analysis of the target and draft distributions of more than 100K tokens across three model families, we approximate this lower bound using only the entropy information of the draft model in Section 2.3, making it viable in actual inference.

2.1 Preliminaries on Speculative Decoding

Suppose we have two LLMs p and q, where p is the larger (target) model, and q is the smaller (draft) model. Given an input sequence $x_{\leqslant t}$ of length t, and a draft length γ, the draft model first samples γ tokens $x_{t+1}, \cdots, x_{t+\gamma}$ autoregressive, which are verified by the target models in parallel to acquire the confidences $p(x_{t+1}), \cdots, p(x_{t+\gamma})$[1].

Then, each draft token x_{n+j} is accepted with probability $\frac{p(x_{n+j})}{q(x_{n+j})}$, and otherwise rejected. In the latter case, a corrected token is sampled from the residual distribution $\frac{\max(q(x_{n+j})-p(x_{n+j}),0)}{\sum_i \max(q(x^i_{n+j})-p(x^i_{n+j}),0)}$, which guarantees that the overall output distribution is exactly the same as $p(x_{n+j})$ (Leviathan et al., 2023; Chen et al., 2023). This process is repeated until a maximum sequence length T is reached.

The complete algorithms for speculative decoding are given in Appendix A.

2.2 Theoretical Lower Bound of Acceptance Rate

From Section 2.1, it's easy to derive that given an input sequence $x_{<t}$ and a draft token x_t, its acceptance probability is $\min\left(1, \frac{p(x_t)}{q(x_t)}\right)$. Let β denote the expected acceptance probability over the distribution of x_t, and it follows that

$$\beta = \sum_x q(x) \cdot \min\left(1, \frac{p(x)}{q(x)}\right) \tag{1}$$

$$= \sum_x \min\left(p(x), q(x)\right). \tag{2}$$

Chen et al. (2023) has proven that β is related to the total variational distance (TVD) between p and q:

$$\beta = 1 - \mathrm{TVD}(p, q). \tag{3}$$

According to Pinsker's inequality - which relates TVD to Kullback-Leibler divergence - we then have

$$\beta \geqslant 1 - \sqrt{\frac{1}{2}\mathrm{KL}(q||p)} \tag{4}$$

$$= 1 - \sqrt{\frac{1}{2}\sum_x q(x)\log\frac{q(x)}{p(x)}}$$

$$= 1 - \sqrt{\frac{1}{2}H_{q,p} - \frac{1}{2}H_q}, \tag{5}$$

where $H_{q,p}$ is the cross entropy between q and p, and H_q is the entropy of q.

Equation (5) provides a theoretical lower bound for the acceptance rate of a draft token using the entropy information of the speculative decoding system[2]. To provide an intuitive motivation for using entropy to construct the lower bound, we plot the relation between draft model entropy and draft token acceptance probability as well as accepted draft seqeunce lengths in Figure 2, which are collected from more than 1M tokens generated by three different target models with temperature set to 1 and maximum draft length set to 40. The first subfigure indicates a strong negative correlation between draft model entropy and draft token

[1]The short hand $p(x_n)$ is used to denote the conditional probability $p(x_n|x_{<n})$ when there is no ambiguity. Throughout the work we use subscripts to indicate token indices in a sequence (e.g. x_n for n-th token), and superscripts to indicate element indices in a vector (e.g. x^i_n for i-th element in x_n).

[2]An alternative way to construct the lower bound is discussed in Appendix B.

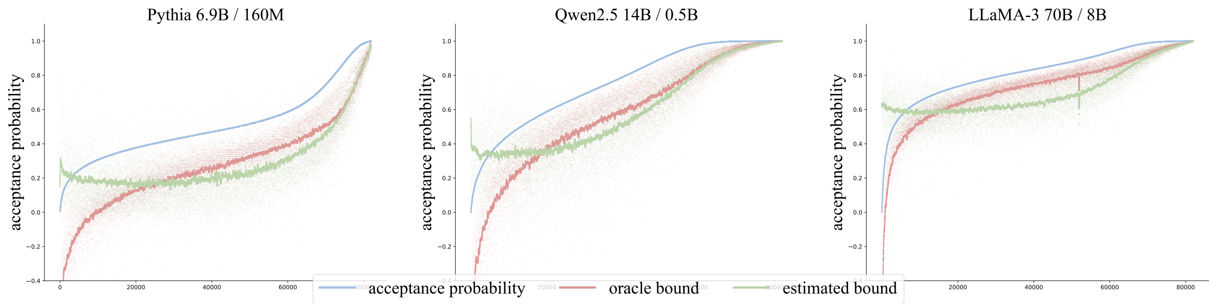

Figure 4: Comparison between the actual acceptance probability from Equation (2), the acceptance probability lower bound from Equation (5), and the estimated lower bound after approximating the cross entropy $H_{q,p}$ with a constant multiplication of H_q. Each position on the x-axis corresponds to a token, which has been sorted according to the actual acceptance probability.

acceptance probability, while the second subfigure shows that when the entropy is low, dozens of consecutive draft tokens could be accepted, which highlights the drawback of setting draft length to a constant, small value, as is the common practice in speculative decoding literature. Another version of Figure 2 in the greedy setting is given in Appendix D.

2.3 Empirical Estimation of Acceptance Rate

So far in this section, we have been assuming access to the target distribution $p(x_t)$ of the next token, which is unavailable in the drafting phase at inference time. Thus, to apply Equation (5) for actual acceptance rate estimation, we must approximate the cross entropy $H_{q,p}$ with information from only the draft model.

To tackle this issue, we first plot the relationship between $H_{q,p}$ and H_q in Figure 3. For all three model families, the entropy ratio $H_{q,p}/H_q$ are concentrated in a narrow range between 1.0 and 1.3. Thus, we choose to approximate $H_{q,p}$ with a constant multiplication of H_q. Plugging it into Equation (5), we now have

$$\beta \geqslant 1 - \sqrt{cH_q}, \tag{6}$$

where c is a constant controlling the approximation ratio between $H_{q,p}$ and H_q. In Figure 4, we visualize the values derived from Equation (2), (5), and (6).

With Equation (6) providing a way to estimate acceptance probability using only the draft model's entropy, we can now adapt the draft length on-the-fly. After generating each draft token, we compute the estimated acceptance probability lower bound, and stop the draft process if it's lower than a certain threshold h. We note that since both c and h are constant hyperparameters, we can remove $\sqrt{c}$ from Equation 6 and absorb it into the threshold h.

Algorithm 1 SVIP

Input: target model p, draft model q, input sequence $x_{\leqslant t}$, maximum length T, threshold h

Initialize $n \leftarrow t$
while $n < T$ **do**
  $j = 0$
  while True **do**
    Sample $x_{n+j} \sim q(x|x_{<n+j})$
    $j \leftarrow j + 1$
    if $\sqrt{H(q_{x|x_{<n+j}})} > h$ **then**
      Exit while loop
    end if
  end while
  $\gamma \leftarrow j$
  Compute $p(x|x_{<n+j}),\ j = 1, \cdots, \gamma + 1$ in parallel
  $\tilde{n} \leftarrow n$
  for $j = 1$ to γ **do**
    if Verify$\left(p_{x|x_{<n+j}}, q_{x|x_{<n+j}}, x_{n+j}\right)$ **then**
      $\tilde{n} \leftarrow \tilde{n} + 1$
    else
      $x_{n+j} \leftarrow$ Correct $\left(p_{x|x_{<n+j}}, q_{x|x_{<n+j}}\right)$
      Exit for loop
    end if
  end for
  if $\tilde{n} == n + \gamma$ **then**
    Sample $x_{n+\gamma+1}$ from $p(x|x_{\leqslant n+\gamma})$
  end if
  $n \leftarrow \tilde{n} + 1$
end while

Output: $x_{\leqslant n}$

We formalize SVIP in Algorithm 1. The details of the methods `Verify` and `Correct` are given in Appendix A, for which different versions are available for sampling (Algorithm 2, 4) and greedy decoding (Algorithm 3, 5).

3 Experiments

3.1 Experiments on SpecBench

3.1.1 Settings

We validate the effectiveness of SVIP on SpecBench (Xia et al., 2024) using three distinct target models: Pythia 6.9B (Biderman et al., 2023), Qwen2.5 14B (Yang et al., 2024), and LLaMA-3 70B (Dubey et al., 2024), with Pythia-160M, Qwen2.5 0.5B, and LLaMA-3 8B as the draft models respectively.

As baselines, we consider two simple policies for draft length: 1) a constant draft length of 5, which is commonly used in the literature, and 2) the heuristics implemented in Hugging Face Transformers library (Wolf et al., 2019), where the draft length for the next draft iteration is increased by 2 if all draft tokens in the current iteration are accepted, and otherwise decreased by 1.

We set the sampling temperature to 0 on SpecBench (the alternatives are discussed in Appendix C). For each model, the entropy threshold t in SVIP is chosen from $\{0.2, 0.3, 0.4, 0.5\}$ based on performance on 8 samples held out from MT-Bench (Zheng et al., 2023), which are 0.4 for Pythia, and 0.3 for Qwen2.5 and LLaMA-3. All experiments with Pythia and Qwen are conducted on a single 40GB A100, while experiments with LLaMA are conducted on 5 40GB A100s. To mitigate the impact of system performance varitations, we repeat all experiments with Pythia and Qwen for three times (using different random seeds when they are used) and report the average speedup over target-model-only autoregressive decoding. Also, since the memory consumption of verifying n draft tokens is quadratic in n, we limit the maximum draft length to 40 in both heuristics and SVIP scenarios, beyond which we start to encounter out-of-memory issues.

3.1.2 Results

The results on SpecBench are shown in Table 1. Compared with the constant approach, SVIP yields an average speedup of 15% for Pythia and 20% for Qwen and LLaMA. Compared with the heuristic approach, SVIP also gives a consistent improvement on all domains for Pythia and Qwen, and outperforms the latter on 4 out of 6 domains for LLaMA.

In Figure 5, we plot the average draft length and accepted draft length of Qwen and LLaMA (the results for Pythia are similar, and are given in Appendix C). From the figure, we observe that by terminating the draft process when the draft model entropy is high, SVIP leads to shorter draft lengths and a much higher acceptance rate (close to or more than 80% on all domains for Qwen, and more than 90% for LLaMA). Notably, the acceptance rate of LLaMA-3 even reaches 99% on the summarization domain, which contributes to the highest speedup (3.48) in Table 1.

3.2 Long-form Generation

Most existing works on speculative decoding (Chen et al., 2023; Du et al., 2024) limit their experiments to generating short sequences of 128 tokens. To verify the wide applicability of SVIP, we also conduct experiments on long-form generation with up to 8K context. For this purpose, we use MT-Bench (Zheng et al., 2023) as the dataset and set the sampling temperature to 1, as we found that when using greedy decoding in long-form generation, both the draft and the target models are prone to repeat themselves, resulting in very low information entropy (see Appendix D for details). Other settings follow Section 3.1.

The results are given in Table 2. Interestingly, we find that in the sampling setting, the two baseline methods (constant and heuristics) perform even worse than target-model-only auto-regressive decoding for Pythia and Qwen, while SVIP consistently yields a positive speedup. For LLaMA, while constant draft length performs better with contexts shorter than 1K tokens, SVIP exceeds it for generating longer sequences. Another observation is that the speedup ratio of all three methods generally increases with the context length, which could be possibly attributed to the longer contexts giving the draft model more information, aligning it at test time to the target model. However, we note that the absolute values of token throughput (which are not discussed in this paper, since they heavily depend on the underlying machines) stay at the same level from 1K to 8K context for Qwen2.5 and LLaMA-3 when using speculative decoding, and decrease slowly when not using speculative decoding. For Pythia, which does not use any optimized attention such as MQA (Shazeer, 2019) or GQA (Shazeer, 2019), the throughput decreases notably with context length in both cases - with or without specualtive decoding.

	Methods	MT-Bench	Trans.	Sum.	QA	Math	RAG	Avg.
Pythia (6.9B, 160M)	Const.	1.45	1.47	1.24	1.43	1.52	1.42	1.42
	Heuristics	1.51	1.58	1.34	1.58	1.64	1.51	1.53
	SVIP	**1.63**	**1.62**	**1.45**	**1.67**	**1.72**	**1.66**	**1.63**(+14.8%)
Qwen2.5 (14B, 0.5B)	Const.	1.08	0.87	1.11	0.92	1.43	0.99	1.07
	Heuristics	1.10	0.91	1.10	0.92	1.34	1.03	1.07
	SVIP	**1.33**	**1.12**	**1.37**	**1.14**	**1.57**	**1.23**	**1.29**(+20.6%)
LLaMA-3 (70B, 8B)	Const.	2.04	2.48	2.56	2.34	2.32	2.28	2.34
	Heuristics	2.30	**3.13**	3.33	2.61	2.52	**2.63**	2.76
	SVIP	**2.31**	3.04	**3.48**	**2.63**	**2.89**	2.59	**2.83**(+20.9%)

Table 1: Speedup over target-model-only autoregressive decoding on SpecBench.

	Methods	Generation Length							
		128	256	512	1K	2K	4K	6K	8K
Pythia (6.9B, 160M)	Const.	0.68	0.69	0.69	0.70	0.72	0.90	0.89	0.87
	Heuristics	0.88	0.88	0.88	0.88	0.90	1.21	1.25	1.23
	SVIP	**1.07**	**1.08**	**1.08**	**1.07**	**1.08**	**1.43**	**1.44**	**1.41**
Qwen2.5 (14B, 0.5B)	Const.	0.98	0.97	0.95	0.96	0.98	1.00	1.02	1.04
	Heuristics	1.01	0.99	0.98	1.00	1.02	1.03	1.04	1.06
	SVIP	**1.29**	**1.29**	**1.30**	**1.31**	**1.32**	**1.33**	**1.33**	**1.35**
LLaMA-3 (70B, 8B)	Const.	**1.74**	**1.74**	**1.77**	**1.78**	1.82	1.90	1.93	1.94
	Heuristics	1.53	1.56	1.61	1.63	1.68	1.77	1.83	1.84
	SVIP	1.69	1.72	1.75	**1.78**	**1.86**	**1.96**	**2.01**	**2.02**

Table 2: Speedup on MT-Bench with different generation length.

3.3 Applying SVIP to Other Draft Methods

In Section 3.1 and 3.2, we evaluated SVIP on vanilla speculative decoding, where a standard pre-trained Transformer decoder model from the target model's family is used as the draft model. However, in the past years many works on speculative decoding have proposed other stronger or more efficient draft models (Cai et al., 2024; Du et al., 2024; Li et al., 2024b). Since most of these works assume a constant draft length, SVIP is orthogonal to them and can be applied on top of them without any additional training.

Specifically, we consider GliDe with a CaPE (Du et al., 2024), where the draft model - named GliDe - is a small transformer decoder with cross-attention to the target model's hidden representations, while CaPE is a complementary tree expansion method to increase the acceptance rate of the draft token at each position. Following the settings of Du et al. (2024), we use Vicuna 7B, 13B, and 33B (Chiang et al., 2023) as the base models (for which the draft models are publicly available) and set the sampling temperature to 0. However, we distribute the 33B model across two GPUs due to memory constraint. Similar to the previous experiments, we set the threshold t to 0.5 based on pilot experiments on 8 samples held out from MT-Bench using GliDe only (without CaPE).

We also apply SVIP to EAGLE-2 (Li et al., 2024a), the state-of-the-art speculative decoding system which utilizes the target model's language modeling head on top of the draft model's features to predict the next draft token, and dynamically constructs a draft tree at each draft position. Following Li et al. (2024a), we use LLaMA-2 7B, 13B (Touvron et al., 2023) and Vicuna 7B, 13B (Chiang et al., 2023) as the base models, and set the sampling temperature to 1. As Brown et al. (2024) suggest, adding a conditional clause (that decides whether or not to stop drafting) after the generation of each draft token in EAGLE-2 may interrupt the otherwise input-independent control flow of

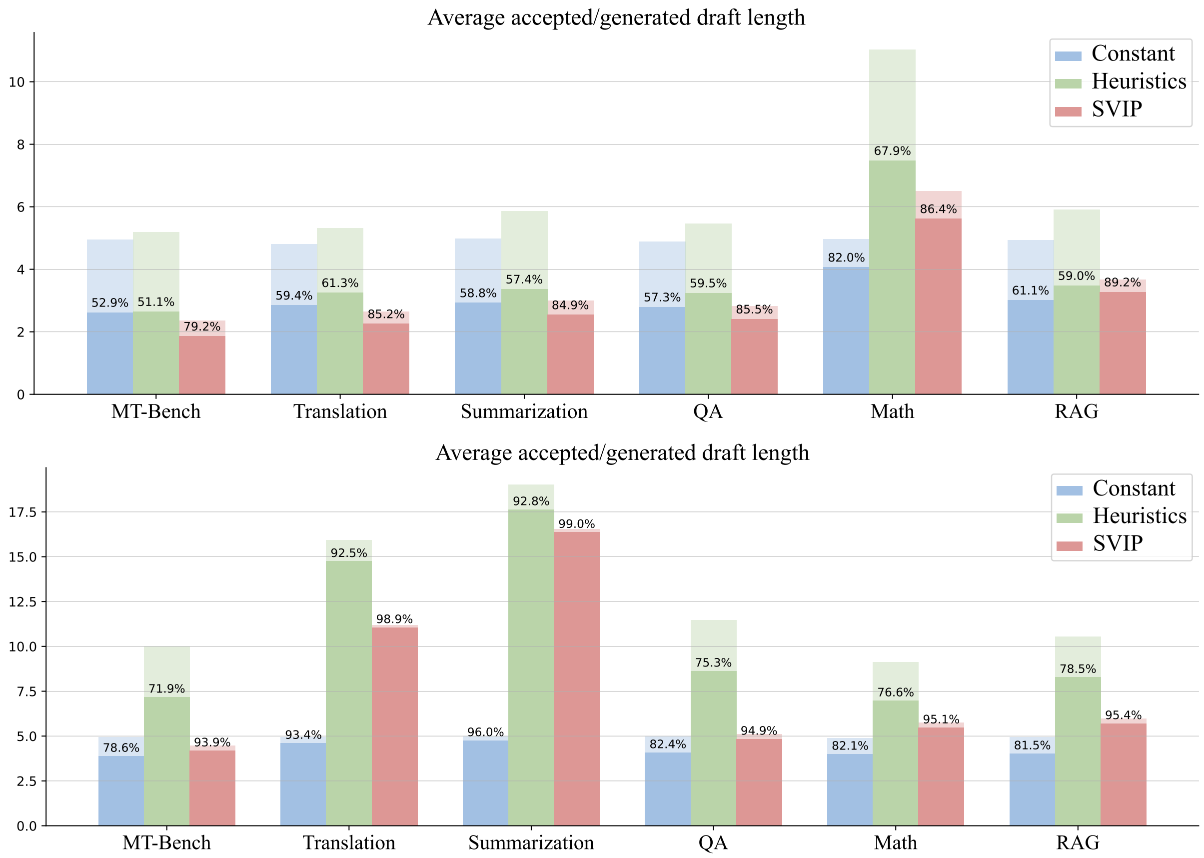

Figure 5: The average generated draft length, accepted draft length, and acceptance rate of Qwen2.5 (top) and LLaMA-3 (bottom) on SpecBench. Compared with the two baselines, SVIP leads to a shorter draft length and a much higher acceptance rate.

EALGE-2, reducing the effects of low-level interpreter and system optimizations. Thus, in EAGLE-2 we calculate the draft entropy and decide whether or not to stop drafting after every two draft tokens instead of after every draft token, as used in previous experiments.

The results for these two methods are presented in Table 3 and 4, respectively. For GliDe, SVIP yields a consistent speedup both with and without CaPE, with a 5% improvement for the 7B and 33B models. For EAGLE-2, the speedup is also particularly notable for the Vicuna models, with 5% improvement for both the 7B and 13B models.

4 Related Work

Since Leviathan et al. (2023) and Chen et al. (2023) introduced speculative decoding into large language models, numerous works have followed their tracks in pursuit of more efficient LLM inference. We broadly categorize these works into three types: better draft models, draft tree expansion, and draft length control, which are orthogonal to each other. A more comprehensive review of speculative decoding is provided by Xia et al. (2024).

Better draft models. As Xia et al. (2024) suggest, draft models in speculative decoding can be either based on self-drafting or based on an independent draft model. For the first type, one may use a quantized (Zhao et al., 2024), early-exiting (Elhoushi et al., 2024), or forward-padded (Monea et al., 2023) version of the target model to produce draft tokens, while the second type is represented by the vanilla speculative decoding (Leviathan et al., 2023). Some works also take the best of both worlds and introduce extra layers on top of the target model's hidden representations to construct draft models, represented by EAGLE (Li et al., 2024b), GliDe (Du et al., 2024), and Medusa (Cai et al., 2024).

Draft tree expansion. Given a draft model, one may verify multiple draft tokens for the same position in parallel to increase the probability of finding an accepted draft token, and we use "draft tree expansion" as an umbrella term for such techniques. Li et al. (2024a) introduce EAGLE-2,

	Methods	MT-Bench	Code	Finance	GSM	Spider	Avg.
7B	GliDe	1.95	2.04	1.91	1.98	1.69	1.95
	+SVIP	2.00	2.12	2.03	2.01	1.63	2.02
	GliDe + CaPE	2.36	2.57	2.29	2.51	1.97	2.40
	+SVIP	**2.56**	**2.65**	**2.49**	**2.54**	**2.08**	**2.52**
13B	GliDe	2.22	2.41	2.15	2.31	1.85	2.24
	+SVIP	2.31	2.43	2.17	2.35	1.85	2.28
	GliDe + CaPE	**2.73**	2.86	**2.66**	2.80	2.24	2.73
	+SVIP	2.72	**2.93**	**2.66**	**2.85**	**2.27**	**2.76**
33B	GliDe	2.12	2.25	2.09	2.29	1.99	2.18
	+SVIP	**2.29**	**2.40**	**2.20**	**2.42**	**2.03**	**2.30**
	GliDe + CaPE	2.08	1.98	2.10	2.13	1.76	2.03
	+SVIP	**2.13**	**2.02**	**2.15**	**2.16**	**1.82**	**2.08**

Table 3: Speedup comparison with GliDe & CaPE, using Vicuna as the base model.

	Methods	MT-Bench	H-Eval	GSM8K	Alpaca	CNN/DM	QA	Avg.
LLaMA-2 7B	EAGLE-2	3.10	3.61	3.15	3.10	2.76	2.84	3.10
	+ SVIP	**3.16**	**3.66**	**3.18**	**3.13**	**2.82**	**3.02**	**3.16**
LLaMA-2 13B	EAGLE-2	3.38	**4.12**	3.41	3.25	3.01	2.98	3.41
	+ SVIP	**3.41**	4.09	**3.45**	**3.34**	**3.05**	**3.17**	**3.46**
Vicuna 7B	EAGLE-2	2.66	2.84	**2.77**	2.48	2.31	2.13	2.58
	+ SVIP	**2.84**	**2.97**	2.75	**2.66**	**2.42**	**2.30**	**2.73**
Vicuna 13B	EAGLE-2	2.85	3.31	2.93	2.74	2.48	2.30	2.83
	+ SVIP	**2.94**	**3.49**	**3.19**	**2.89**	**2.60**	**2.53**	**2.98**

Table 4: Speedup comparison with EAGLE-2, using LLaMA-2-Chat and Vicuna as the base models.

which reranks draft tokens in EAGLE's draft tree to select tokens with the highest confidence for verification. Similarly, CaPE (Du et al., 2024) improves GliDe by expanding the token set chosen for verification at each position based on top-1 confidence. Other works have also addressed the problem of multi-draft verification from a theoretic perspective (Sun et al., 2023; Yin et al., 2024).

Draft length control. Works in this category are few, but most relevant to ours. Liu et al. (2024) introduce PEARL, which lets the target model perform verification in parallel to draft generation, stopping the draft process when a mismatch is found. Huang et al. (2024) propose SpecDec++, which trains an acceptance prediction head on top of the draft model to predict the acceptance probability of the current draft token, stopping the draft round when the predicted acceptance probability falls below a constant threshold. Brown et al. (2024) propose Dynamic Depth Decoding on top of EAGLE-2, which uses the sum of all tokens' confidences in one level of its draft tree as an indicator to predict whether or not to continue draft generation.

5 Conclusion

We propose SVIP, a flexible, training-free, and plug-and-play dynamic draft length policy for speculative decoding systems. Based on a theoretical lower bound of acceptance probability and its empirical approximation, SVIP determines whether to continue draft generation or to quit drafting based on the draft model's entropy after the generation of each draft token. With extensive experiments spanning various base models, draft methods, test domains, and generation length, we validated the effectiveness of SVIP, sparking new insights on speculative decoding and more efficient large language models.

References

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 2397–2430. PMLR.

Jean Bretagnolle and Catherine Huber. 1978. Estimation des densités : risque minimax. *Séminaire de probabilités de Strasbourg*, 12:342–363.

Oscar Brown, Zhengjie Wang, Andrea Do, Nikhil Mathew, and Cheng Yu. 2024. Dynamic depth decoding: Faster speculative decoding for llms. *CoRR*, abs/2409.00142.

Tianle Cai, Yuhong Li, Zhengyang Geng, Hongwu Peng, Jason D. Lee, Deming Chen, and Tri Dao. 2024. Medusa: Simple LLM inference acceleration framework with multiple decoding heads. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. 2023. Accelerating large language model decoding with speculative sampling. *CoRR*, abs/2302.01318.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Cunxiao Du, Jing Jiang, Yuanchen Xu, Jiawei Wu, Sicheng Yu, Yongqi Li, Shenggui Li, Kai Xu, Liqiang Nie, Zhaopeng Tu, and Yang You. 2024. Glide with a cape: A low-hassle method to accelerate speculative decoding. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco Guzmán, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vítor Albiero, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. 2024. The llama 3 herd of models. *CoRR*, abs/2407.21783.

Mostafa Elhoushi, Akshat Shrivastava, Diana Liskovich, Basil Hosmer, Bram Wasti, Liangzhen Lai, Anas Mahmoud, Bilge Acun, Saurabh Agarwal, Ahmed Roman, Ahmed A Aly, Beidi Chen, and Carole-Jean Wu. 2024. Layerskip: Enabling early exit inference and self-speculative decoding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 12622–12642. Association for Computational Linguistics.

Kaixuan Huang, Xudong Guo, and Mengdi Wang. 2024. Specdec++: Boosting speculative decoding via adaptive candidate lengths. *CoRR*, abs/2405.19715.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. 2023. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 19274–19286. PMLR.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. 2024a. EAGLE-2: faster inference of language models with dynamic draft trees. *CoRR*, abs/2406.16858.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. 2024b. EAGLE: speculative sampling requires rethinking feature uncertainty. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Tianyu Liu, Yun Li, Qitan Lv, Kai Liu, Jianchen Zhu, and Winston Hu. 2024. Parallel speculative decoding with adaptive draft length. *CoRR*, abs/2408.11850.

Xiaofan Lu, Yixiao Zeng, Feiyang Ma, Zixu Yu, and Marco Levorato. 2024. Improving multi-candidate speculative decoding. *CoRR*, abs/2409.10644.

Giovanni Monea, Armand Joulin, and Edouard Grave. 2023. Pass: Parallel speculative sampling. *CoRR*, abs/2311.13581.

Noam Shazeer. 2019. Fast transformer decoding: One write-head is all you need. *CoRR*, abs/1911.02150.

Ziteng Sun, Ananda Theertha Suresh, Jae Hun Ro, Ahmad Beirami, Himanshu Jain, and Felix X. Yu. 2023. Spectr: Fast speculative decoding via optimal transport. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

Heming Xia, Zhe Yang, Qingxiu Dong, Peiyi Wang, Yongqi Li, Tao Ge, Tianyu Liu, Wenjie Li, and Zhifang Sui. 2024. Unlocking efficiency in large language model inference: A comprehensive survey of speculative decoding. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 7655–7671. Association for Computational Linguistics.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. 2024. Qwen2 technical report. *CoRR*, abs/2407.10671.

Ming Yin, Minshuo Chen, Kaixuan Huang, and Mengdi Wang. 2024. A theoretical perspective for speculative decoding algorithm. *CoRR*, arXiv:2411.00841.

Juntao Zhao, Wenhao Lu, Sheng Wang, Lingpeng Kong, and Chuan Wu. 2024. Qspec: Speculative decoding with complementary quantization schemes. *CoRR*, abs/2410.11305.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

A The Complete Speculative Decoding Algorithms

In Algorithm 2 to 6, we present the complete algorithms of the vanilla speculative decoding in both the greedy decoding and the sampling scenarios. For the sampling scenario, the `Verify` and `Correct` methods in Algorithm 6 resolve to Algorithm 2 and 4. For greedy decoding, they resolve to Algorithm 3 and 5.

Algorithm 2 Verify (Sampling)

Input: target distribution $p(x)$, draft distribution $q(x)$, draft token x_t

$accept \leftarrow$ False
$r \sim U[0, 1]$
if $r < \frac{p(x_t)}{q(x_t)}$ **then**
  $accept \leftarrow$ True
end if

Output: $accept$

Algorithm 3 Verify (Greedy)

Input: target distribution $p(x)$, draft distribution $q(x)$, draft token x_t

$accept \leftarrow$ False
if $\arg\max p(x) == x_t$ **then**
  $accept \leftarrow$ True
end if

Output: $accept$

Algorithm 4 Correct (Sampling)

Input: target distribution $p(x)$, draft distribution $q(x)$

Sample $\hat{x} \sim \frac{\max(q(x)-p(x),0)}{\sum_i \max(q(x^i)-p(x^i),0)}$

Output: $\hat{x}$

Algorithm 5 Correct (Greedy)

Input: target distribution $p(x)$, draft distribution $q(x)$

Output: $\arg\max p(x)$

Algorithm 6 Speculative Decoding

Input: target model p, draft model q, input sequence $x_{\leqslant t}$, maximum length T, draft length γ

Initialize $n \leftarrow t$
while $n < T$ **do**
  for $j = 1$ to γ **do**
    Sample $x_{n+j} \sim q(x|x_{<n+j})$
  end for
  Compute $p(x|x_{<n+j}),\ j = 1, \cdots, \gamma + 1$ in parallel
  $\tilde{n} \leftarrow n$
  for $j = 1$ to γ **do**
    if Verify$(p(x|x_{<n+j}),\ q(x|x_{<n+j}), x_{n+j})$ **then**
      $\tilde{n} \leftarrow \tilde{n} + 1$
    else
      $x_{n+j} \leftarrow$ Correct $(p(x|x_{<n+j}),\ q(x|x_{<n+j}))$
      Exit for loop
    end if
  end for
  if $\tilde{n} == n + \gamma$ **then**
    $x_{n+\gamma+1} \sim p(x|x_{\leqslant n+\gamma})$
  end if
  $n \leftarrow \tilde{n} + 1$
end while

Output: $x_{\leqslant n}$

B Alternatives for Acceptance Rate Lower Bound Computation

In Section 2, we used Pinsker's inequality to compute a lower bound for the expected acceptance probability:

$$\beta = \sum_x \min\left(p(x), q(x)\right) \tag{7}$$

$$\geqslant 1 - \sqrt{\frac{1}{2} KL(q||p)}. \tag{8}$$

Another way to compute the lower bound of acceptance probability can be derived from Bretagnolle-Huber inequality (Bretagnolle and Huber, 1978):

$$\beta \geqslant 1 - \sqrt{1 - e^{-KL(q||p)}}. \tag{9}$$

Compared with the Pinsker's bound, it's trivial to see that this bound is guaranteed to be always larger than 0. However, in practice we find that the Pinsker's bound is 11% tighter for Qwen2.5, 20% tighter for Pythia, and 43% tighter for LLaMA-3.

C Additional Results on SpecBench

In Figure 6, we plot the draft length and acceptance rate of Pythia (which complements Figure 5 in Section 3.1), and in Figure 7 we also give the results of the three models when using sampling instead of

	Methods	MT-Bench	Trans.	Sum.	QA	Math	RAG	Avg.
Pythia 6.9B, 160M	Const.	0.65	0.63	0.65	0.66	0.65	0.64	0.65
	Heuristics	0.82	0.83	0.85	0.83	0.83	0.83	0.83
	SVIP	**1.05**	**1.02**	**1.03**	**1.01**	**1.03**	**1.00**	**1.02** $_{(+56.9\%)}$
Qwen2.5 14B, 0.5B	Const.	1.01	0.85	0.87	0.85	1.32	0.86	0.96
	Heuristics	1.02	0.94	0.93	0.88	1.22	0.91	0.99
	SVIP	**1.24**	**1.08**	**1.19**	**1.11**	**1.47**	**1.10**	**1.20** $_{(+25.0\%)}$
LLaMA-3 70B, 8B	Const.	**1.62**	**1.56**	1.65	**1.53**	**1.73**	1.54	**1.60**
	Heuristics	1.56	1.55	**1.76**	1.49	1.61	1.55	1.58
	SVIP	1.53	1.53	1.69	1.51	1.71	**1.56**	1.58 $_{(-1.3\%)}$

Table 5: Speedup on SpecBench using temperature sampling.

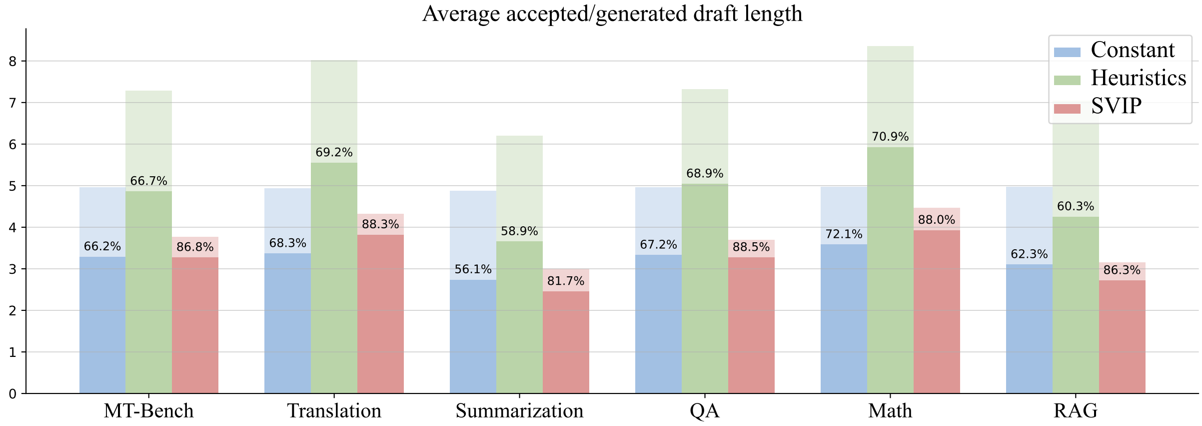

Figure 6: The average generated draft length, accepted draft length, and acceptance rate of Pythia on SpecBench.

greedy decoding[3]. Across the different models and sampling methods, the same observation as discussed in Section 3 holds: SVIP results in short draft lengths and a much higher acceptance rate.

D Additional Results on Long-form Generation

In Table 6, we present the results of long-form generation with greedy decoding. Compared with Table 2, the speedup ratio of greedy decoding is much higher (even more than 4 times for LLaMA-3 after 4K context). This is due to the fact that these models tend to repeat themselves in greedy long-form generation, making it very easy for the draft models to predict the next tokens. In Figure 8, we also plot the relation between draft model entropy and accepted draft sequence lengths in this setting, which shows a much stronger correlation compared with the sampling setting in Figure 2.

[3]We note that in the "constant" scenario, the draft length is always set to 5. However, if an EOS token is sampled from the draft model, the draft process will terminate. So the overall draft length might be slightly lower than 5, as can be seen from Figure 7.

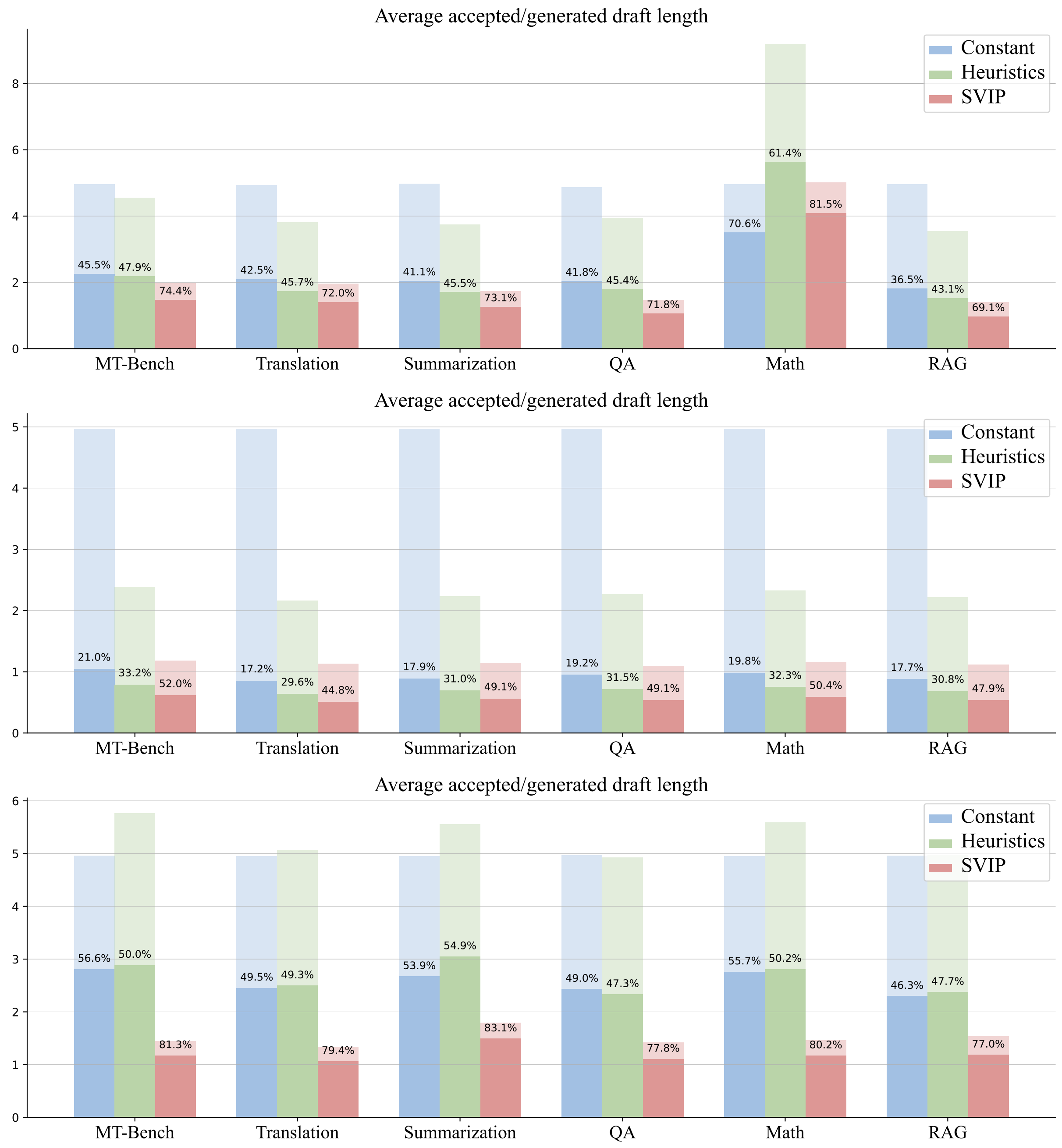

Figure 7: The average generated draft length, accepted draft length, and acceptance rate of Qwen2.5 (top), Pythia (middle), and LLaMA-3 (bottom) on SpecBench, using temperature sampling instead of greedy decoding.

	Methods	Generation Length							
		128	256	512	1K	2K	4K	6K	8K
Pythia (6.9B, 160M)	Const.	1.10	1.30	1.50	1.66	1.82	1.20	1.11	1.04
	Heuristics	1.25	1.45	1.65	1.81	2.02	1.50	1.46	1.41
	SVIP	**1.41**	**1.62**	**1.83**	**2.01**	**2.21**	**1.71**	**1.60**	**1.52**
Qwen2.5 (14B, 0.5B)	Const.	1.05	1.08	1.15	1.29	1.44	1.54	1.60	1.67
	Heuristics	1.04	1.06	1.13	1.32	1.54	1.72	1.85	1.97
	SVIP	**1.30**	**1.34**	**1.42**	**1.57**	**1.74**	**1.87**	**1.98**	**2.10**
LLaMA-3 (70B, 8B)	Const.	2.06	2.18	2.31	2.45	2.58	2.72	2.77	2.78
	Heuristics	2.26	2.46	2.73	3.07	3.48	3.90	4.15	4.26
	SVIP	**2.31**	**2.56**	**2.86**	**3.21**	**3.59**	**4.00**	**4.23**	**4.33**

Table 6: Speedup on MT-Bench with different generation length using greedy decoding.

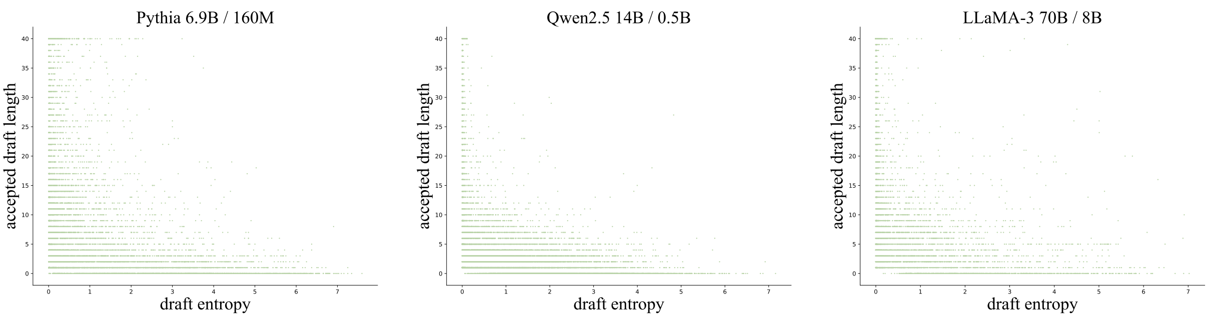

Figure 8: The correlation between draft model entropy and lengths of accepted draft seqeunces in the greedy setting.